

**AMENDMENTS TO THE CLAIMS:**

Please add new claim 50, and amend claims 1, 7, 12, 15, 17, 18, 24, 29, 32 to 35, 40, 42, 43, 45, 47 and 48, as shown below. This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for transforming application data structures into an XML document, the method comprising:

[[a)] writing an application program having Advanced Business Application Programming ("ABAP") data structures;

[[b)] writing a transformation program;

[[c)] executing the application program;

[[d)] executing the transformation program when called for by the application program to transform the ABAP data structures from the application program into an XML document; and

[[e)] sending the XML document to a recipient.

2. (Original) The method of claim 1, wherein executing the transformation program further comprises executing the transformation program on a dedicated virtual machine.

3. (Original) The method of claim 2, wherein the virtual machine runs on a web application server.

4. (Original) The method of claim 1, wherein writing the transformation program further comprises compiling the transformation program into a byte-code language.

5. (Original) The method of claim 4, wherein executing the transformation program further comprises executing the compiled transformation program on a dedicated virtual machine.

6. (Original) The method of claim 1, wherein the transformation program is written in markup language syntax.
7. (Currently Amended) The method of claim 1, wherein the transformation program includes a construct for reading a value within the ABAP data structures and writing the value to the XML document.
8. (Original) The method of claim 1, wherein the transformation program allows for literal XML elements, attributes, and text that appear within the transformation program to be written to the XML document.
9. (Original) The method of claim 1, wherein the transformation program includes a construct for specifying attributes to be written to the XML document.
10. (Original) The method of claim 1, wherein the transformation program includes a construct for declaring namespaces in the XML document.
11. (Original) The method of claim 1, wherein the transformation program includes a construct for skipping program instructions.
12. (Currently Amended) The method of claim 1, wherein the transformation program includes a construct for copying elements from ABAP data structures to the XML document.
13. (Original) The method of claim 1, wherein the transformation program includes a construct for calling another transformation program.
14. (Original) The method of claim 1, wherein the transformation program includes a construct for applying a transformation template associated with another transformation program called by the application program.

15. (Currently Amended) The method of claim 1, wherein the transformation program includes a construct for looping over ABAP data structures while creating the XML document.
16. (Original) The method of claim 1, wherein the transformation program includes a construct for executing conditional logic to create certain XML content within the XML document.
17. (Currently Amended) The method of claim 1, wherein the transformation program used to convert the ABAP data structures into the XML document can be used to convert the XML document back into ABAP data structures.
18. (Currently Amended) A method for transforming an XML document into application data structures, the method comprising:
  - [(a)] writing an application program configured to use Advanced Business Application Programming ("ABAP") data structures;
  - [(b)] writing a transformation program;
  - [(c)] executing the application program;
  - [(d)] executing the transformation program when called for by the application program to transform an XML document into ABAP data structures for the application program; and
  - [(e)] using the ABAP data structures within the application program.
19. (Original) The method of claim 18, wherein executing the transformation program further comprises executing the transformation program on a dedicated virtual machine.
20. (Original) The method of claim 19, wherein the virtual machine runs on a web application server.
21. (Original) The method of claim 18, wherein writing the transformation program further comprises compiling the transformation program into a byte-code language.

22. (Original) The method of claim 21, wherein executing the transformation program further comprises running the compiled transformation program on a dedicated virtual machine.
23. (Original) The method of claim 18, wherein the transformation program is written in a markup language syntax.
24. (Currently Amended) The method of claim 18, wherein the transformation program includes a construct for reading a value within the XML document and writing the value to the ABAP data structures.
25. (Original) The method of claim 18, wherein the transformation program includes literal XML elements that are matched in the XML document.
26. (Original) The method of claim 18, wherein the transformation program includes a construct for matching the name of an attribute in the XML document.
27. (Original) The method of claim 18, wherein the transformation program includes a construct for matching a namespace declaration in the XML document.
28. (Original) The method of claim 18, wherein the transformation program includes a construct for skipping program instructions.
29. (Currently Amended) The method of claim 18, wherein the transformation program includes a construct for copying elements to the ABAP data structures from the XML document.
30. (Original) The method of claim 18, wherein the transformation program includes a construct for calling another transformation program.

31. (Original) The method of claim 18, wherein the transformation program includes a construct for applying a transformation template associated with another transformation program called by the application program.

32. (Currently Amended) The method of claim 18, wherein the transformation program includes a construct for looping over content from the XML document while creating the ABAP data structures.

33. (Currently Amended) The method of claim 18, wherein the transformation program includes a construct for executing conditional logic to create certain ABAP data structures.

34. (Currently Amended) The method of claim 18, wherein the transformation program used to convert the XML document into ABAP data structures can be used to convert the ABAP data structures back into the XML document.

35. (Currently Amended) An application system comprising:

a first process configured to execute an application program, wherein the application program is operable to use a set of Advanced Business Application Programming ("ABAP") data structures;

a second process configured to interpret a markup language document;

a transformation template configured to specify a symmetric mapping between the markup language document and the set of ABAP data structures; [[and]]

a transformation virtual machine running in association with the first process and operable to execute the transformation template, [[;]] wherein the transformation virtual machine is operable to:

perform a symmetric transformation between the markup language document and the set of ABAP data structures to allow the first process and the second process to exchange information [[.]]; and

provide the transformed data to the application program.

36. (Original) The application system of claim 35 wherein the first process is an ABAP virtual machine running on an application server.
37. (Original) The application system of claim 35 wherein the second process is one of a client processor and a server processor configured to communicate with the application server.
38. (Original) The application system of claim 36 further comprising a database configured to communicate with the application server, wherein the database is operable to store at least one of XML data and non-XML data.
39. (Original) The application system of claim 36 wherein the application server is a web application server.
40. (Currently Amended) A method for enabling application data structures to be used at multiple times by one or more application programs, the method comprising:  
    executing an application program, the application program using one or more Advanced Business Application Programming ("ABAP") data structures;  
    executing a transformation program when called for by the application program, the transformation program being configured to [(1)] transform the one or more ABAP data structures into extensible markup language (XML) content, the XML content including one or more XML structures corresponding to the one or more ABAP data structures, and [(2)] transform the XML structures of the XML content into the one or more ABAP data structures;  
    and  
    providing the transformed data structures to the application program.
41. (Previously Presented) The method of claim 40 further comprising storing the XML content in a data store from which the XML content may be later retrieved.
42. (Currently Amended) The method of claim 41 further comprising:  
    accessing the XML content from the data store;

executing the transformation program to transform the accessed XML content into the one or more ABAP data structures used by the application program; and  
executing the application program using the one or more ABAP data structures.

43. (Currently Amended) The method of claim 40 further comprising transmitting the XML content to an external location for transformation into a second set of ABAP data structures to be used by a second application program through execution of a second transformation program that is configured to transform the transmitted XML content into the second set of ABAP data structures.

44. (Previously Presented) The method of claim 40 further comprising compiling the transformation program in preparation for the execution of the transformation program on a dedicated virtual machine.

45. (Currently Amended) A method for enabling application data structures to be used at multiple times by one or more application programs, the method comprising:

executing an application program, the application program using one or more Advanced Business Application Programming ("ABAP") data structures;

executing a transformation program when called for by the application program, the transformation program being configured to [(1)] transform the one or more ABAP data structures into extensible markup language (XML) data, the XML data including one or more XML structures corresponding to the one or more ABAP data structures, and [(2)] transform the XML structures of the XML data into the one or more ABAP data structures; and  
providing the transformed data structures to the application program.

46. (Previously Presented) The method of claim 45 further comprising storing the XML data in a data store from which the XML data may be later retrieved.

47. (Currently Amended) The method of claim 46 further comprising:  
accessing the XML data from the data store;

executing the transformation program to transform the accessed XML data into the one or more ABAP data structures used by the application program; and

executing the application program using the one or more ABAP data structures.

48. (Currently Amended) The method of claim 45 further comprising transmitting the XML data to an external location for transformation into a second set of ABAP data structures to be used by a second application program through execution of a second transformation program that is configured to transform the transmitted XML data into the second set of ABAP data structures.

49. (Previously Presented) The method of claim 45 further comprising compiling the transformation program in preparation for the execution of the transformation program on a dedicated virtual machine.

50. (New) A method for transforming application data structures into an XML document, the method comprising:

writing an application program having non-XML data structures;

writing a transformation program;

executing the application program;

executing the transformation program when called for by the application program to transform the non-XML data structures from the application program into an XML document;  
and

sending the XML document to a recipient.